

ALTERNATIVE ROUTING IN PNNI HIERARCHICAL NETWORKS

2 FIELD OF THE INVENTION

3 This invention relates to alternative routing of connections between source and destination
4 nodes in PNNI hierarchical networks.

5 BACKGROUND

6 PNNI (Private Network-to-Network Interface) is a standards-based signaling and routing
7 protocol which was approved by the ATM (Asynchronous Transfer Mode) Forum in 1996 for use in
8 ATM systems. The PNNI protocol provides, inter alia, a system for creation and distribution of
9 topology information which determines how individual network nodes "see" the network and thus
10 how nodes communicate. A key feature of the protocol is the ability to cluster groups of switches
11 into "peer groups". The details of each peer group are abstracted into a single logical node which is
12 all that can be seen outside of that peer group. This system is applied recursively so that PNNI can
13 hierarchically aggregate network topology information. The hierarchical system is illustrated
14 schematically in Figure 1 of the accompanying drawings which shows a simple, three-level
15 hierarchical network. At the lowest level of the hierarchy in this example, the nodes labeled A1 to
16 D2 represent real switches. The switches A1, A2 and A3 are clustered to form a peer group which is
17 represented by a logical node A in the next level up of the hierarchy. The switches B1 and B2 form
18 a peer group which is represented by logical node B in the next level. Similarly, switches C1 to C3
19 and switches D1 and D2 are represented by nodes C and D respectively in the next level. Further,
20 nodes A and B are clustered and represented by logical node X in the upper level of the hierarchy.
21 Nodes C and D are also clustered and are represented by logical node Y in the upper level.

22 The topology information available to the switches is such that each switch sees the details
23 of its own peer group plus the details of any peer group that represents it at a higher level of the
24 PNNI hierarchy. Thus, the network structure as seen by switch A1 in the above example is as
25 illustrated in Figure 2 of the accompanying drawings. Switch A1 sees two links to logical node B
26 and one link from B to logical node Y. Node B represents all of the switches in that group, namely

1 B1 and B2, and node Y represents all of the switches in groups C and D, namely C1, C2, C3, D1
2 and D2. As demonstrated by this Figure, the hierarchical aggregation of topology information leads
3 to loss of information. For example, the details of switches C1 to D2 are hidden to switch A1 which
4 sees only the logical node Y representing these switches. This loss of information can lead to
5 problems with the routing of connections for communication between nodes as will be
6 demonstrated below.

7 ATM is a source routing technology. When a connection is established between nodes, the
8 route that the connection takes through the network is specified by the first switch. This switch
9 specifies the route as a series of paths, one for each level of the PNNI hierarchy that is traversed
10 between the source and destination. The series of paths defining a route is known as a DTL
11 (Designated Transit List). For example, referring again to Figures 1 and 2, suppose that a call is
12 received at switch A1 addressed to an end system attached to switch D2. Due to the hierarchy,
13 switch A1 does not see D2 but rather the node Y which represents D2 at a higher level. Thus, A1
14 may compute the following DTL for establishment of the connection:

15 X, Y
16 A, B
17 A1, A2

18 At the lowest level of the hierarchy the call goes from A1 to A2. At the second level of the hierarchy
19 the call goes from A to B. At the top level of the hierarchy the call goes from X to the destination at
20 Y. In the real network, the connection setup will start at A1 and be forwarded to A2 as specified.
21 Switch A2 will forward the setup to switch B1 since it knows that its neighbor is in group B which
22 is specified in the computed path. Switch B1 will forward the setup to switch C1 since it knows
23 that its neighbor is in group Y which is specified in the computed path. C1 determines from the call
24 address that the call should be forwarded to node D, and thus from C1 the call may be transferred to
25 C2, and from there via D1 to the destination node D2.

26 To increase the probability that a connection can be established successfully, the ATM
27 Forum defines a crankback mechanism. If a call setup fails, the call is returned to node(s) that
28 created the path to attempt to choose an alternative path ("alternative routing"). To achieve this, the

19 The weakness of the above system can be seen from consideration of Figure 1 which shows
20 that there is a route over which the call could be established successfully, namely: A1, A3, B2, C3,
21 C2, D1 and D2. Thus, if switch A1 had originally specified the route:

24 A1, A3

3

1 SUMMARY

2 According to one aspect of the present invention there is provided a method for alternative
3 routing of a connection between a source node and a destination node in a PNNI hierarchical
4 network, the method comprising responding to a failed connection between said nodes due to a
5 sole-access element of the network structure as seen by the source node, where a said sole-access
6 element is an element which provides sole access to the destination node in said network structure,
7 by:

8 selecting at least one non-sole-access element of the route used by the failed connection in
9 said network structure;

10 identifying an alternative route for the connection in said network structure which does not
11 utilize the at least one selected element; and

12 using the alternative route for establishment of the connection between said nodes.

13 DESCRIPTION OF THE FIGURES

14 Preferred embodiments of the invention are described, by way of example, with reference to
15 the accompanying drawings in which:

16 Figure 1 is a schematic representation of an example of a PNNI hierarchical network;

17 Figure 2 illustrates the network structure as seen by the switch A1 in Figure 1;

18 Figure 3 is a schematic representation of another example of a PNNI hierarchical network;

19 and

20 Figure 4 illustrates the network structure as seen by the switch A1 in Figure 3.

21 DESCRIPTION

22 The present invention is predicated on the realization that a failed connection due to a
23 sole-access element of the network structure seen by the source node can occur due to selection of
24 another element in the route. Thus, by altering the selection of one or more non-sole-access
25 elements (ie. elements which are not sole-access elements), a route by which the connection can be
26 established correctly can often be found. For example, in the scenario discussed above where the

1 connection failed on the basis that the sole-access node Y was blocked, a different result could have
2 been achieved by making a different selection of the non-sole access elements earlier in the route.
3 Evidently if there are no non-sole access elements in a particular failed route then there will be no
4 alternative route to try. However, where there is at least one non-sole access element there will be at
5 least one alternative route, and embodiments of the present invention can identify this and reroute
6 the connection accordingly. Thus, embodiments of the present invention allow successful rerouting
7 of connections where the previous rerouting scheme would have failed, improving the overall
8 performance and robustness of networks in which the invention is applied.

9 In preferred embodiments of the invention, the method includes the step of checking whether
10 the alternative route satisfies a set of predefined connection constraints, wherein the alternative
11 route is used for establishment of the connection only if said constraints are satisfied. The set of
12 connection constraints may include one or more constraints and these may vary for different
13 connections. In particular, PNNI supports the use of Quality of Service (QoS) parameters whereby
14 incoming calls can specify certain constraints which the network must satisfy when establishing the
15 connection. Typical constraints relate, for example, to the bandwidth required by a call and the
16 acceptable delay introduced by the connection. The PNNI protocol allows a source node to
17 determine if these constraints can be met by a particular route, and if not the route will be rejected.

18 While methods can be envisaged in which the selected element or elements of the failed
19 route are nodes of the network structure, it is preferred that the selected elements, ie. the elements
20 which are avoided when determining the alternative route, are links of the network structure.
21 Various systems can be used for selecting the particular elements to be avoided when determining
22 an alternative route, and examples of preferred systems will be described in detail below.

23 In methods embodying the invention, if a successful connection is not established with a
24 first-identified alternative route (eg. because the route does not meet any specified connection
25 constraints or because the connection fails again when the alternative route is used), then the
26 method may be applied again making a different selection of non-sole-access elements if available,
27 to identify a different alternative route. Thus, methods embodying the invention may be iterative.
28 The number of attempts which are made to find a successful route may of course be limited to avoid

1 excessive use of resources. For example, while multiple attempts may be made to identify a "valid
2 route" (ie. a route which meets any specified connection constraints), the number of attempts to
3 reroute a connection using different valid routes may be limited. In particular, some embodiments
4 may make only one attempt at rerouting a connection in order to preserve network resources.

5 It is to be understood that, in general, where features are described herein with reference to a
6 method embodying the invention, corresponding features may be provided in accordance with
7 apparatus embodying the invention, and vice versa. In particular, a further aspect of the present
8 invention provides apparatus for alternative routing of a connection between a source node and a
9 destination node in a PNNI hierarchical network, the apparatus comprising:

10 memory for storing topology data, defining the network structure as seen by the source node,
11 and route data indicative of a route in said network structure used for establishment of a connection
12 between the source node and a destination node;

13 control logic configured to respond to a failed connection between said nodes due to a
14 sole-access element of the network structure as seen by the source node, where a said sole-access
15 element is an element which provides sole access to the destination node in said network structure,
16 by:

17 selecting at least one non-sole-access element of the route used by the failed connection in
18 accordance with said route data;

19 identifying from said topology data an alternative route for the connection which does not
20 utilize the at least one selected element; and

21 outputting the alternative route for establishment of the connection between said nodes.

22 The apparatus may be embodied in a source node such as a switch, router, bridge, brouter or
23 other network device. Alternatively, the apparatus may be embodied in a dedicated route server
24 associated with a peer group including the source node. The invention also extends to a PNNI
25 hierarchical network comprising such apparatus.

26 Preferred embodiments of the invention will now be described, by way of example, with
27 reference to the accompanying drawings.

1 In the following, a preferred method embodying the invention will initially be described in
2 detail, and operation of the preferred method will be described in relation to the networks of Figures
3 1 and 3. For the purpose of this example, it will be assumed that each switch in the exemplary
4 networks is capable of implementing the rerouting method, so that each switch in its own right
5 constitutes apparatus embodying the invention. To enable a switch to implement the alternative
6 routing method, control logic for implementing the various steps to be described is provided in the
7 switch. The control logic may be implemented in hardware or software or a combination thereof,
8 and suitable implementations will be apparent to those skilled in the art from the following
9 description.

10 The rerouting method performed by a switch can be summarized conveniently by the
11 following pseudo-code, where it is assumed initially that a failed connection between that switch
12 and a destination node has just occurred.

13 1. Block all the links used by the failed connection that are outside our peer group;
14 2. if (any link is a sole-access link)
15 then do
16 unblock the link giving access to it;
17 done

18 3. Store the list of blocked links in BL;
19 Sort BL by decreasing distance from source node (links closest to the destination at the
20 top);

21 4. Compute the route;
22 if (route is valid)

1 then do
2 give the route to signalling;
3 End;
4 done

5 5. Unblock all the links;

6 6. for all the elements in BL do
7 Unblock previous element if there was one;
8 Block the current element;
9 Compute the route;
10 if (route is valid)
11 then do
12 Give the route to signalling;
13 End;
14 done
15 done

16 7. Give 'No route to destination' to signalling.

17 In the above algorithm, a number of the steps refer to blocking or unblocking of certain
18 links. In reality, the links are not of course physically blocked or unblocked, but rather certain links
19 are selected or deselected, and as a result of this selection/deselection process certain links will be
20 considered to be blocked for the purpose of identifying an alternative route. This will be apparent
21 from the following description in which the way that the algorithm is implemented by a source
22 switch is considered in more detail.

23 In step 1, the switch control logic initially selects ("blocks") all links of the route used by the
24 failed connection which are outside its own peer group. The links used by the failed connection are

1 defined by route data which has previously been stored in a memory of the switch and can be
2 accessed by the control logic when required for the alternative routing scheme. In step 2, the control
3 logic deselects ("unblocks") any sole-access links since these must be used by any route to the
4 destination node. The control logic can identify sole-access links by reference to the topology data
5 which has been set up previously in accordance with the PNNI protocol and which is stored in the
6 switch memory. This topology data defines the network structure seen by the switch as described
7 earlier. In step 3, the control logic stores the remaining selected links, (the "blocked links"), as a
8 blocked-link list BL in the switch memory. The control logic then sorts the links in BL in order of
9 decreasing distance from the source node such that the link which is closest to the destination node
10 is at the top of the list.

11 After this initial selection process, in step 4 the control logic analyzes the topology data with
12 reference to the list BL to determine if there is an alternative route to the destination which does not
13 utilize any of the "blocked" links in BL. If an alternative route is identified, the control logic then
14 checks, by means of the known processing operations specified by PNNI which need not be detailed
15 here, whether the route satisfies any connection constraints, such as QoS bandwidth or delay
16 requirements etc., which have been specified by the incoming call. Again, these constraints will be
17 stored in the switch memory for access by the control logic. If the constraints are met then the
18 alternative route is deemed to be valid. If the route is valid, the control logic outputs the route to the
19 signaling circuitry of the switch (not described but of known form) for establishing the connection,
20 and the process is finished. If, on the other hand an alternative route identified in this step does not
21 satisfy the connection constraints and is therefore determined to be invalid, the process proceeds to
22 step 5 wherein all the links are "unblocked". Again, this "unblocking" of the links in step 5 is only
23 notional in that the "blocked" links in BL are simply treated by the control logic as deselected, or
24 "unblocked", for the start of step 6 to which the process now proceeds.

25 Step 6 begins with an iterative process in which the control logic selects, or "blocks", each
26 link in BL in turn, starting at the top of the list. When a link is selected, the control logic then
27 determines as described above whether there is a valid alternative route which does not utilize the
28 selected link. If so the route is output to the signaling circuitry for rerouting the connection and the

1 process terminates. If there is no valid route for the currently-selected link then the process proceeds
2 to the next pass of the iteration in which the currently-selected link is deselected, the next link in the
3 list BL is selected, and a valid alternative route is sought which does not utilize the newly-selected
4 link. If, after all links in BL have been selected in turn, no valid route has been found, then the
5 process proceeds to step 7 wherein the control logic outputs a message indicating that no valid route
6 to the destination node is available. This message is communicated to the signaling circuitry for
7 transfer to the device which originated the call.

8 Consideration of the above rerouting algorithm shows that it can be divided into two parts.
9 The first part consists of steps 1 to 4. This part aims to identify the most disjoint (most different)
10 route to that used by the failed connection. This is done by selecting all links in the set of
11 non-sole-access links in the route used by the failed connection which are outside the peer group of
12 the source switch, and identifying an alternative route which does not utilize any of these links. If
13 this fails then the second part, consisting of steps 5 to 7, is implemented. This part involves
14 selecting the non-sole access links of the aforementioned set one at a time, starting with the link
15 closest to the destination node, and checking for an alternative route which does not utilize the
16 selected link.

17 This particular algorithm makes only one attempt at rerouting a failed connection. Thus,
18 when a valid alternative route has been used for establishing a connection, if the connection fails
19 again then no attempt is made to find another alternative route. The first part of the method looks for
20 the most disjoint route since this generally has the best chance of success if used for rerouting the
21 connection. In the second part of the method, selecting links in order of increasing distance from the
22 destination node makes it likely that a more disjoint path, which generally provides a better chance
23 of a successful connection than a less disjoint path, will be found first. Thus, while only one
24 rerouting attempt is made to keep use of network resources to a minimum, the method ensures that
25 there is a good chance that this one attempt will be successful.

26 To demonstrate operation of the rerouting method, consider first the situation previously
27 described with reference to Figure 1 where a failed connection between source node A1 and

1 destination node D2 has just occurred due to insufficient bandwidth in the links from C1 to C2 and
2 C1 to C3 with the following route:

3 X, Y
4 A, B
5 A1, A2

6 In this case, switch A1 is a switch embodying the invention which can implement the preferred
7 method as described above. Thus, when the call is returned to A1 with the cause code "Node Y is
8 blocked", the rerouting algorithm is implemented even though node Y is a sole-access element. In
9 accordance with step 1 of the method, A1 first "blocks" all links used by the failed connection
10 outside of its peer group, namely A2 - B, and B - Y. It is apparent from Figure 2, however, that the
11 link B - Y is a sole-access link so this link is "unblocked" in step 2. The list BL in step 3 therefore
12 contains only A2 - B. The alternative route identified in step 4 is therefore:

13 X, Y
14 A, B
15 A1, A3

16 Assuming this route satisfies any specified connection constraints and is therefore deemed valid, the
17 route will be output for establishment of the connection. In the real network, the route will cause the
18 connection to go through the following sequence of switches: A1, A3, B2, C3, C2, D1 and D2, and
19 the connection will be established successfully.

20 Consider next the two-level hierarchical network of Figure 3. The lowest level of the
21 hierarchy in this network is similar to that of Figure 1 except that there are additional switches C4
22 and D3 interconnected as shown. The four groups of switches A1 to A3, B1 and B2, C1 to C4, and
23 D1 to D3 are clustered to form peer groups which are represented by logical nodes A, B, C and D in
24 the upper level of the hierarchy as indicated. In this example, however, the links interconnecting
25 peer groups are not aggregated in the hierarchy, so that, in the upper level, successive nodes A, B, C
26 and D are interconnected by two links. In each pair of links interconnecting two nodes, the upper
27 link in each pair is labeled 1 in the figure indicating that this link originates at a first port (port 1) of
28 the preceding node. The lower link of each pair is labeled 2 in the figure indicating that this link

1 originates at a second port (port 2) of the preceding node. The network structure as seen by node A1
2 in this case is as shown in Figure 4.

3 Suppose that, as before, switch A1 has received a call addressed to an end system attached to
4 D2 and computes the following initial route for the connection:

5 A1, B1, C1, D

6 A1, A2

7 where, in the path A1, B1, C1, D, the notation "A1" signifies node A, port 1, "B1" signifies node B,
8 port 1, and so on, so that the upper links 1 in Figure 4 should be used. In the real network the call
9 setup is forwarded by A1 to A2 as specified in the route. Switch A2 then forwards the setup directly
10 to B1 via the upper link 1 as specified in the route. Similarly, B1 forwards the setup directly to C1
11 via the upper link 1 as specified in the route. Suppose that, as before, the links from C1 to C2 and
12 from C1 to C3 do not have sufficient bandwidth to support the call so that the call cannot be
13 completed. The crankback mechanism is then used to return the call to the source switch A1 with
14 the cause code "Node C is blocked". Node C is a sole-access element of the network structure as
15 seen by A1 since any connection to the destination node for the call must utilize node C. Thus, with
16 the existing rerouting scheme specified by PNNI, A1 would not attempt to reroute the call since a
17 sole-access element is blocked. Here, however, the switches A1 to D2 are configured for
18 implementing the rerouting algorithm described in detail above. Thus, when the call is returned to
19 A1, A1 will initially block the links A1 - B, B1 - C and C1 - D used by the failed connection outside
20 its peer group (step 1). None of these links are sole-access links (step 2), so that, at the end of step 3,
21 the blocked link list BL will contain all these links in the order C1 - D, B1 - C, A1 - B. In step 4, A1
22 computes the following route which uses none of the "blocked" links in B:

23 A2, B2, C2, D

24 A1, A3

25 where, in the path A2, B2, C2, D, the notation "A2" signifies node A, port 2, "B2" signifies node B,
26 port 2, and so on, so that the lower links 2 in Figure 4 should be used. For the sake of example,
27 assume now that this route does not meet the specified delay constraints for the call, so that the
28 route is not valid. All the links in BL are then "unblocked" (step 5) and the process proceeds to step

1 6. In the first stage of step 6, the first link in the list BL, ie. C1 - D is initially blocked and the
2 control logic of A1 computes an alternative route which does not utilize this link. Since there is
3 more than one possible route here, in this embodiment the control logic selects a particular route at
4 random, though of course other selection systems could be used if desired. Thus, for example, A1
5 may identify the following route:

6 A1, B2, C2, D

7 A1, A2

8 Again for the sake of example, suppose that this route is deemed invalid. Switch A1 will then
9 continue the search for a valid route by selecting another route which avoids the "blocked" link C1 -
10 D, for example using the upper-level path A2, B1, C2, D. Suppose further that this route, and each
11 of the other possible routes which do not use the link C1 - D, is also found to be invalid. Switch A1
12 then continues with step 6 by unblocking C1 - D and blocking the next link in BL, namely B1 - C.
13 A1 now computes a route which does not utilize B1 - C. Again, since there is more than one route to
14 choose from here, a particular route will be selected at random. This may be a route already
15 identified as an invalid route, eg. the route using the upper level path A1, B2, C2, D, so the control
16 logic will select another route, eg:

17 A2, B2, C1, D

18 A1, A3.

19 Assuming this route satisfies the connection constraints, the route will be output to signaling. In the
20 real network, the connection will pass through the following sequence of switches: A1, A3, B2, C3,
21 C2, D1 and D2, and the connection will be established successfully.

22 When considering the application of the preferred algorithm in the networks of Figures 1 and
23 3, the description has focused on examples of failed connections where the failure is due to a
24 sole-access element of the network structure as seen by the source node since these are the situations
25 where the existing rerouting scheme would inevitably fail to find an alternative route. It will be
26 appreciated however, that in practice the algorithm will be applied in response to any failed

1 connection whether or not due to a sole-access element, and provides a highly efficient rerouting
2 system in all circumstances.

3 It will also be appreciated that, while a particularly preferred embodiment has been
4 described in detail above, many changes and modifications can be made to the embodiment
5 described without departing from the scope of the invention. For example, in some situations it may
6 be acceptable, or even preferable, for step 3 of the algorithm to sort the links in BL by increasing
7 distance from the source node, ie links closest to the source node at the top. In some embodiments,
8 therefore, the control logic may be configured to implement the algorithm with this modification in
9 all cases, or the control logic may be configured to select which of the two orders to adopt in a
10 particular situation. In addition, in some embodiments step 6 of the above algorithm may be
11 modified by removal of the statement "unblock previous element if there was one". The effect of
12 this would be that, when performing step 6, the control logic would successively block more links in
13 each pass of the iterative process. This would reduce processing time by restricting the number of
14 routes tried by the control logic, while still providing a good chance of successfully rerouting a
15 connection.

16 As a further example, some embodiments may utilize only one part of the algorithm
17 described above, ie. steps 1 to 4 or steps 5 and 6. While the two-part algorithm is clearly more likely
18 to result in successful rerouting, use of either part on its own will still allow successful rerouting to
19 be achieved in many cases where the prior scheme would have failed, and these simpler algorithms
20 may be appropriate in some circumstances, eg. where it is desired to simplify the processing as far
21 as possible.

22 Further, while the above method makes only one attempt at rerouting a connection, in some
23 embodiments it may be desirable to make more than one attempt, or even to try all possible valid
24 routes before returning a "No route to destination" code. In these cases, if a valid alternative route is
25 used for establishing a connection and the connection fails again, then the failed route may be stored
26 in the control logic as an invalid route, and the method repeated to identify a new alternative route.

27 As a further example, in the networks as described above, each switch incorporates control
28 logic for implementing the rerouting method when a failed call is returned to the switch. In other

1 embodiments, the rerouting method may be implemented by a dedicated node, such as a route
2 server, which manages rerouting of calls for any source node within a particular peer group. Thus,
3 one such route server may be provided for each peer group, the route server being connected in the
4 peer group and sharing the same view of the network topology as the other nodes in the group.
5 When a failed call is returned to a source node in a given peer group, the source node simply sends
6 the details of the failed route to the route server together with any connection constraints and the
7 cause code for the connection failure. The route server then implements the rerouting method as
8 described above, returning the alternative route to the source node for forwarding to signaling.
9 The present invention can be realized in hardware, software, or a combination of hardware and
10 software. The present invention can be realized in a centralized fashion in one computer system, or
11 in a distributed fashion where different elements are spread across several interconnected computer
12 systems. Any kind of computer system - or other apparatus adapted for carrying out the methods
13 described herein - is suitable. A typical combination of hardware and software could be a general
14 purpose computer system with a computer program that, when being loaded and executed, controls
15 the computer system such that it carries out the methods described herein. The present invention
16 can also be embedded in a computer program product, which comprises all the features enabling the
17 implementation of the methods described herein, and which - when loaded in a computer system - is
18 able to carry out these methods.

19 Computer program means or computer program in the present context is meant to include
20 any expression, in any language, code or notation, of a set of instructions intended to cause a system
21 having an information processing capability to perform a particular function either directly or after
22 either or both of the following a) conversion to another language, code or notation; b) reproduction
23 in a different material form.

24 It is noted that the foregoing has outlined some of the more pertinent objects and
25 embodiments of the present invention. Thus, although the description is made for particular
26 arrangements and methods, the intent and concept of the invention is suitable and applicable to
27 other arrangements and applications. It will be clear to those skilled in the art that modifications to
28 the disclosed embodiments can be effected without departing from the spirit and scope of the

1 invention. The described embodiments ought to be construed to be merely illustrative of some of
2 the more prominent features and applications of the invention. Other beneficial results can be
3 realized by applying the disclosed invention in a different manner or modifying the invention in
4 ways known to those familiar with the art.

004080" 292E950